AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method performed by a computer system comprising:

providing a test module for dynamically reinstalling an associated test module interface at anytime without rebooting the computer system, in response to a device change associated with the test module;

in response, unloading the test module associated with the changed device and causing its test module interface to be reinstalled without rebooting the computer system;~~ and~~

using a registration module to register uses of the test module by a plurality of programs so that no other program attempts can access information associated with the test module after it unloads~~.~~;

in response to the test module seeking to unload, accessing the registration module to determine what other programs are registered as using the test module;

notifying each of the programs to stop using the test module;

each program responding that they have stopped using the test module or are still using the test module;

the test module causing itself to unload in response to programs responding that using the test module is stopped; and

the test module cancelling unloading in response to programs responding that the test module is being used, whereby the test module may unload at a later time.

2. (Previously Presented) The method of claim 1, further comprising:

calling a function identified by a test module interface to cause a first test configuration of the test module to be created;

detecting a test routine associated with the test module using the first test configuration; and

causing the test routine to be executed.

3. (Cancelled).

4. (Cancelled).

5. (Previously Presented) The method of claim 2, further comprising:

in response to detecting the change,

calling the function to cause a second test configuration of the test module to be created.

6. (Original) The method of claim 1, further comprising:

registering a use of the test module by a program.

7. (Original) The method of claim 6, further comprising:

unloading the test module; and

informing the program of the unloading prior to unloading the test module.

8. (Original) The method of claim 7, further comprising:

conveying a defer signal from the program to the test module; and

in response to the defer signal, canceling the unloading of the test module.

9. (Currently Amended) A computer system comprising:

a processor;

a memory coupled to the processor;

a test module dynamically reinstalling its test module interface at any time without rebooting the computer system, in response to a device change associated with that test module;

in response, the test module associated with the changed device being unloaded and causing its test module interface to be reinstalled without rebooting the computer system;~~ and~~

a registration module being used to register uses of the test module by a plurality of programs so that no other program attempts to access information associated with the test module after it unloads~~.~~;

means for accessing the registration module to determine what other programs are registered as using the test module;

means for notifying each of the programs to stop using the test module;

means for each program to respond that they have stopped using the test module or are still using the test module;

means for the test module to cause itself to unload in response to programs responding that using the test module is stopped; and

means for the test module to cancel unloading in response to programs responding that the test module is being used, whereby the test module may unload at a later time.

10. (Currently Amended) The computer system of claim 9, wherein the~~ program is executable to~~ system:

~~call~~calls a function identified by a test module interface to cause a first test configuration of the test module to be created;

~~detect~~detects a test routine associated with the test module using the first test configuration; and

~~cause~~causes the test routine to be executed.

11. (Cancelled).

12. (Cancelled).

13. (Previously Presented) The computer system of claim 10, wherein the function identified by the test module interface is called to cause a second test configuration of the test module to be created.

14. (Previously Presented) The computer system of claim 9, wherein a use of the test module by the program is registered.

15. (Previously Presented) The computer system of claim 14, wherein the test module is unloaded; and

the program is notified of the unloading prior to unloading the test module.

16. (Previously Presented) The computer system of claim 15, wherein a defer signal is conveyed from the program to the test module; and

in response to the defer signal, the unloading of the test module is canceled.

17. (Cancelled).

18. (Cancelled).

19. (Cancelled).

20. (Cancelled).

21. (Cancelled).

22. (Cancelled).